SECTION II—REMARKS

Applicants thank the Examiner for a thorough review, and respectfully request reconsideration of the above referenced patent application for the following reasons:

Claims 20-33 rejected under 35 U.S.C. § 101

The Office Action rejected claims 20-33 under 35 U.S.C. § 101 because "the claimed subject matter does not belong to any of the four statutory categories … ." Claims 20-33 are canceled herein without prejudice, and thus, the rejection is rendered moot.

However, Applicants respectfully submit that each of new claims 49-92 are directed toward patentable subject matter, and thus, are in condition for allowance under 35 U.S.C. § 101.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 20-33 and allow new claims 49-92.

Claims 1-20, 22-29, 31-34, 36-38, and 40-48 rejected under 35 U.S.C. § 102(a)

The Office Action rejected claims 1-20, 22-29, 31-34, 36-38, and 40-48 under 35 U.S.C. § 102(a) as being unpatentable over Sun Microsystems, "*Building Web Services - Sun™ ONE Studio 5 Programming Series*," published in June 2003 ("Sun Web Services"). Claims 1-48 are canceled herein without prejudice, and thus, the rejection to claims 1-20, 22-29, 31-34, 36-38 and 40-48 is rendered moot. However, Applicants respectfully submit that new claims 49-92 presented herein, are not anticipated by Sun Web Services. In particular, new independent claim 49, recites in pertinent part:

> obtaining a description of a **Web service comprising protocol-independent business logic**;

generating the Web service …

generating a first virtual interface … comprising a mapping of the protocol-independent business logic … to a first transport protocol …

generating a second virtual interface … comprising a mapping of the protocol-independent business logic … to a second transport protocol different than the first transport protocol … .

Brief description of the claimed limitations:

In an effort to aid in the expeditious prosecution of the present application, Applicants provide a brief description of the claimed limitations.

In the specification as originally filed, Applicants teach a method for enabling business logic of a particular Web service to be accessible to Web service clients operating on different communication and transport protocols, and optionally using different authentication mechanisms.

More particularly, the specification teaches a method of abstracting such protocols from the Web Service, resulting in a Web service with protocol-independent business logic. Through the use of an abstraction layer, called a "virtual interface," the protocol-independent business logic of the Web service may be later mapped to a particular transport protocol suiting the particular needs of a Web service client attempting to communicate with the Web service, without having to regenerate the Web service itself. Because the underlying business logic of the Web service is protocol-independent, a single Web service programmed at "design-time" or "development-time," can be used to service multiple Web service clients having different transport protocol needs, including those determined at "run-time," through the use of multiple virtual interfaces. Applicants teach and claim an embodiment in which each virtual interface has its own transport protocol mapping to underlying protocol-independent business logic of the

Web service, thus enabling the single Web service to communicate with multiple clients communicating over diverse protocols.

Consider for example, the following excerpts of the original specification teaching in pertinent part:

> [**0002**] ... Web services are self-contained, modularized, executable entities that can be published, searched for, and accessed across a network. ... .
>
> * * *
>
> [**0007**] In some cases, it may be advantag[eous] to extend the generated Web service client so that it implements additional functions. For example, the security of service consumer 120 may be enhanced if the Web service client includes **an authentication extension**. Similarly, the processing of message traffic between service consumer 120 and service provider 110 may be enhanced if the Web service client includes a flexible **extension for processing message headers**. Conventional Web service clients do not include these extensions.

Thus, the specification as originally submitted contemplates the ability to extend the functionality of Web service clients, possibly enhancing authentication and message traffic processing. The original specification further teaches that Web services contain "business logic," which provide the functionality or "operations" for a particular Web service:

> [**00011**] ... Web service provider 210 includes **business logic** 212 ... The term "**business logic**" refers to software that performs data processing. Business logic 212 may provide the **operations that are packaged as a Web service**.

The original specification goes on to teach a Web service having protocol-independent business logic rather than conventional protocol specific or protocol-dependent business logic:

> [**00016**] FIG. 4 is a block diagram of the general architecture of **Web service 400** ... .
>
> * * *
>
> [**00019**] In an embodiment, Web service design time part 420 provides a **description of Web service 400 in terms of abstract features**, rather than specific technical implementations. Thus, the developer of Web service design time part 420 may

> **focus on the logic** of Web service implementation 410 **rather than the actual binding information** used to expose Web service 400. … .
>
> * * *
>
> [**00074**] … a feature of the **client protocol implementation** is set to define a behavior of the Web service client **without regenerating the [Web service] client proxy** at 1540. … .
>
> * * *
>
> [**000113**] In an embodiment, **the Web service implementation** (as well as the Web service definition, and the virtual interface) **is independent of the transport layer protocol**.

Thus, Applicants teach a "Web service" that is not dependent on "specific technical implementations," but rather describes functionality in terms of "abstract features," thus allowing a developer to "focus on the logic … rather than the **actual binding information** used to expose [the] Web service." Such a design yields a "Web service implementation … **independent** of the transport layer protocol," that may define various client protocol implementations "without regenerating" the Web service. [Refer to the above paragraphs and Figure 15].

The original specification further teaches a "virtual interface" capable of mapping the "protocol-independent business logic" to a "transport protocol," as claimed. For example:

> [**00024**] An advantage to the architecture of Web service 400 is that **a single implementation** 410 may be **exposed in multiple ways**. … [I]mplementation 410 may have **multiple virtual interfaces** 422. … .
>
> [**00025**] In an embodiment, Web service configuration part 430 **binds an abstract Web service to particular transports, bindings, and protocols**. … .
>
> [**00026**] In an embodiment, Web service configuration 434 specifies which **transport binding** will be used, a **security configuration**, a target address, and/or documentation for the operations of the configuration. In addition, Web service configuration 434 may **specify which design-time feature will be mapped to which runtime feature**. The term "design time" refers to the design and development of computer software. The term "runtime" refers to the actual execution of software. … .
>
> * * *

> [**000113**] … Each configuration may specify, for example, a **transport binding, mapping of abstract design-time features to runtime protocol implementations**, security configuration, target address, an the like. … .

Thus, Applicants teach and claim a "virtual interface … comprising a mapping of the protocol-independent business logic … to a [] transport protocol," and further teach that a "single implementation [e.g., a single Web service] may be exposed in multiple ways" and have "multiple virtual interfaces."

The cited reference does not disclose the claimed limitations:

In its rejection of independent claim 1, the Office Action relies on Sun Web Services, a reference book that, according to its introduction, "explains how [to] build simple and complex [W]eb services using the Sun™ ONE Studio 5, Standard Edition integrated development environment (IDE)." The 214 page book is a generic technical guide regarding Web service technologies as they relate to Sun's IDE, rather than a specific disclosure of novel advancements in Web services technologies.

In rejecting now canceled claim 1, the Office Action references pages 29, 95-105, and 112-122, each of which disclose concepts generally related to Web services, but none of which contemplate or disclose the particular embodiments taught and claimed by Applicants. For example, page 29 describes WSDL documents and UDDI registries, pages 95-105 describe generating, configuring, deploying, and executing JAX-RPC clients, and pages 112-122 generally describe how to create a Web service client from either a WSDL file directly or based on information stored in a UDDI registry.

However, Sun Web Services is silent with respect to multiple virtual interfaces that map different transport protocols to a single Web service. In particular, Sun Web Services fails to disclose:

> ... generating a first virtual interface ... comprising a mapping of the protocol-independent business logic ... **to a first transport protocol** ...
> generating a second virtual interface ... comprising a mapping of the protocol-independent business logic ... **to a second transport protocol different than the first transport protocol** ...
> .

Although Sun Web Services does disclose the use of Simple Object Access Protocol (SOAP) which can, "define[] the mechanism by which a web service is called and how data is returned" (refer to page 27, last paragraph), the reference is silent with respect to "a first virtual interface ... comprising a **mapping** of the protocol-independent business logic ... **to a first transport protocol**," or a second such "virtual interface" having a "mapping of the protocol-independent business logic to a **second transport protocol different than the first transport protocol**," as Applicants recite in new independent claim 49. Indeed, there is no discussion whatsoever of multiple "virtual interfaces" to a single Web service, much less multiple virtual interfaces each "comprising a mapping of the **protocol-independent business logic**" to different "transport protocols," as taught and claimed by Applicants.

Because Sun Web Services fails to disclose at least one limitation claimed by Applicants in amended independent claim 1, Applicants respectfully submit that new independent claim 49 is not anticipated by Sun Web Services and is in condition for allowance. Applicants further submit that new independent claims 59, 71, and 83, which recite similar limitations, as well as those claims dependent on independent claims 49, 59, 71, and 83, are not anticipated by Sun Web Services and are in condition for allowance.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 1-20, 22-29, 31-34, 36-38, and 40-48 canceled herein, and allow new claims 49-92 presented herein.

Dependent claims 30 and 39 rejected under 35 U.S.C. § 103(a)

The Office Action rejected claims 30 and 39 under 35 U.S.C. § 103(a) as being unpatentable over Sun Web Services in view of Sun Microsystems, "Sun One Architecture Guide – *Delivering Services on Demand*," published in 2002 ("Sun DSOD").

Dependent claims 30 and 39 are canceled herein without prejudice, and thus, the rejection of claims 30 and 39 is rendered moot. However, Applicants respectfully submit that Sun DSOD fails to cure the deficiencies of Sun Web Services as it too fails to disclose a "a first virtual interface … comprising a **mapping** of the protocol-independent business logic … **to a first transport protocol**," or a second such "virtual interface" having a "mapping of the protocol-independent business logic to a **second transport protocol different than the first transport protocol**," as Applicants recite in new independent claim 49 and the similar limitations of new claims 50-92.

Because Sun Web Services and Sun DSOD, whether considered alone or in combination, fail to disclose at least one limitation as Applicants recite in new independent claims 49-92, Applicants respectfully submit that new claims 49-92 are patentable over the references and in condition for allowance.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 30 and 39 and allow new claims 49-92.

New claims 49-92:

In accordance with the above remarks in reference to the rejection under 35 U.S.C. § 102, Applicants respectfully submit that new claims 49-92 presented herein are patentable over the prior art of record and in condition for allowance. New claims 49-92 find support in Applicants' original specification and in the original claims submitted with the specification.

Accordingly, Applicants respectfully request the Examiner to allow new claims 49-92.

CONCLUSION

Given the above amendments and accompanying remarks, all claims pending in the application are in condition for allowance. If the undersigned attorney has overlooked subject matter in any of the cited references that is relevant to allowance of the claims, the Examiner is requested to specifically point out where such subject matter may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (503) 439-8778.

Charge Deposit Account

Please charge our Deposit Account No. 02-2666 for any additional fee(s) that may be due in this matter, and please credit the same deposit account for any overpayment.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

/Gregory D. Caldwell/ Date: June 9, 2008
Gregory D. Caldwell
Registration No. 39,926
Attorney for Applicants

Blakely, Sokoloff, Taylor & Zafman LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
Telephone: (503) 439-8778
Facsimile: (503) 439-6073